**Lab 02 report**

**Completed By:** Dominic Choi

**UCID:** 30109955

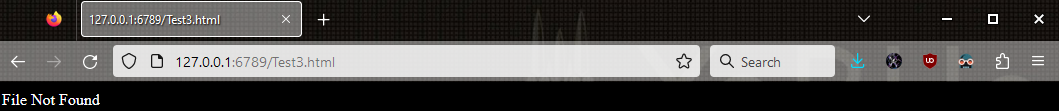
**Part 1**

A close up of a computer screen

Description automatically generated

A screen shot of a computer

Description automatically generated



**Part 2**

Python File:

UDPPingerClient.py

|  |
| --- |
| from socket import \*  import time  # Server address and port  server\_ip = "127.0.0.1"  # Change this to the actual server IP address  server\_port = 12000  # Number of ping messages to send  num\_pings = 10  # Create a UDP socket  client\_socket = socket(AF\_INET, SOCK\_DGRAM)  # Initialize variables for RTT statistics  min\_rtt = float("inf")  max\_rtt = 0  total\_rtt = 0  packets\_lost = 0  # Ping loop  for sequence\_number in range(1, num\_pings + 1):      # Get the current time for timestamp      send\_time = time.time()      # Construct the ping message      ping\_message = f'Ping {sequence\_number} {send\_time}'      try:          # Send the ping message to the server          client\_socket.sendto(ping\_message.encode(), (server\_ip, server\_port))          # Set a timeout for receiving the response          client\_socket.settimeout(1.0)          # Receive the response from the server          response, server\_address = client\_socket.recvfrom(1024)          # Get the current time again for RTT calculation          receive\_time = time.time()          # Calculate the RTT          rtt = receive\_time - send\_time          # Update RTT statistics          min\_rtt = min(min\_rtt, rtt)          max\_rtt = max(max\_rtt, rtt)          total\_rtt += rtt          # Print the response and RTT          print(f'Response from {server\_ip}: {response.decode()} (RTT: {rtt:.6f} seconds)')      except timeout:          # Packet was lost (request timed out)          print(f'Request timed out {sequence\_number}')          packets\_lost += 1  # Calculate packet loss rate  packet\_loss\_rate = (packets\_lost / num\_pings) \* 100  # Print statistics  print(f'\nPing statistics for {server\_ip}:')  print(f'    Packets: Sent = {num\_pings}, Received = {num\_pings - packets\_lost}, Lost = {packets\_lost} ({packet\_loss\_rate:.2f}% loss)')  print(f'Approximate round-trip times in milliseconds:')  print(f'    Minimum = {min\_rtt \* 1000:.6f} ms, Maximum = {max\_rtt \* 1000:.6f} ms, Average = {(total\_rtt / (num\_pings - packets\_lost)) \* 1000:.6f} ms')  # Close the socket  client\_socket.close() |

Output (with UDPPingerServer.py running)

A screen shot of a computer

Description automatically generated